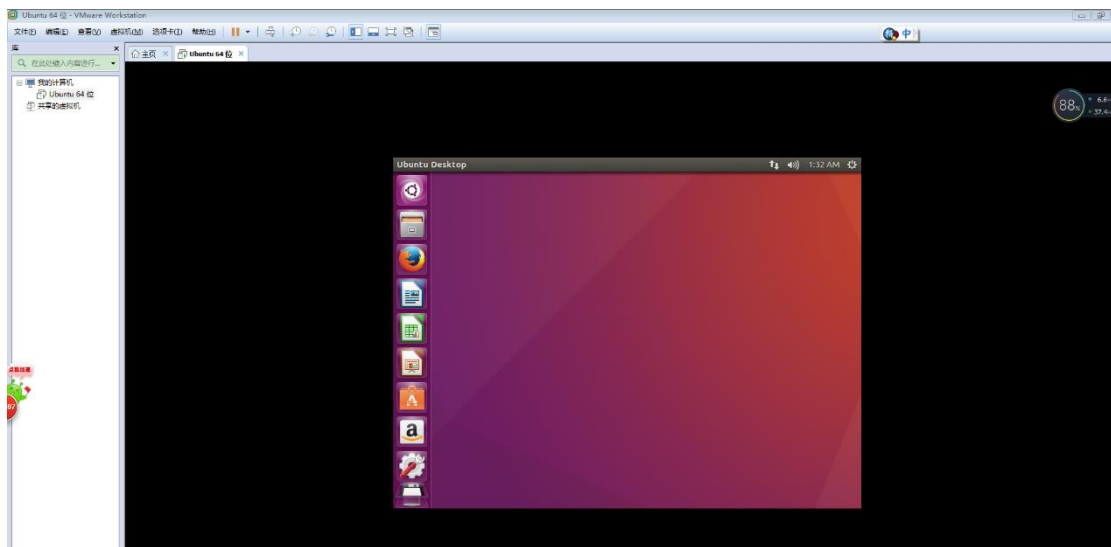


第一章 准备工作

1.1 VMware 中安装 ubuntu

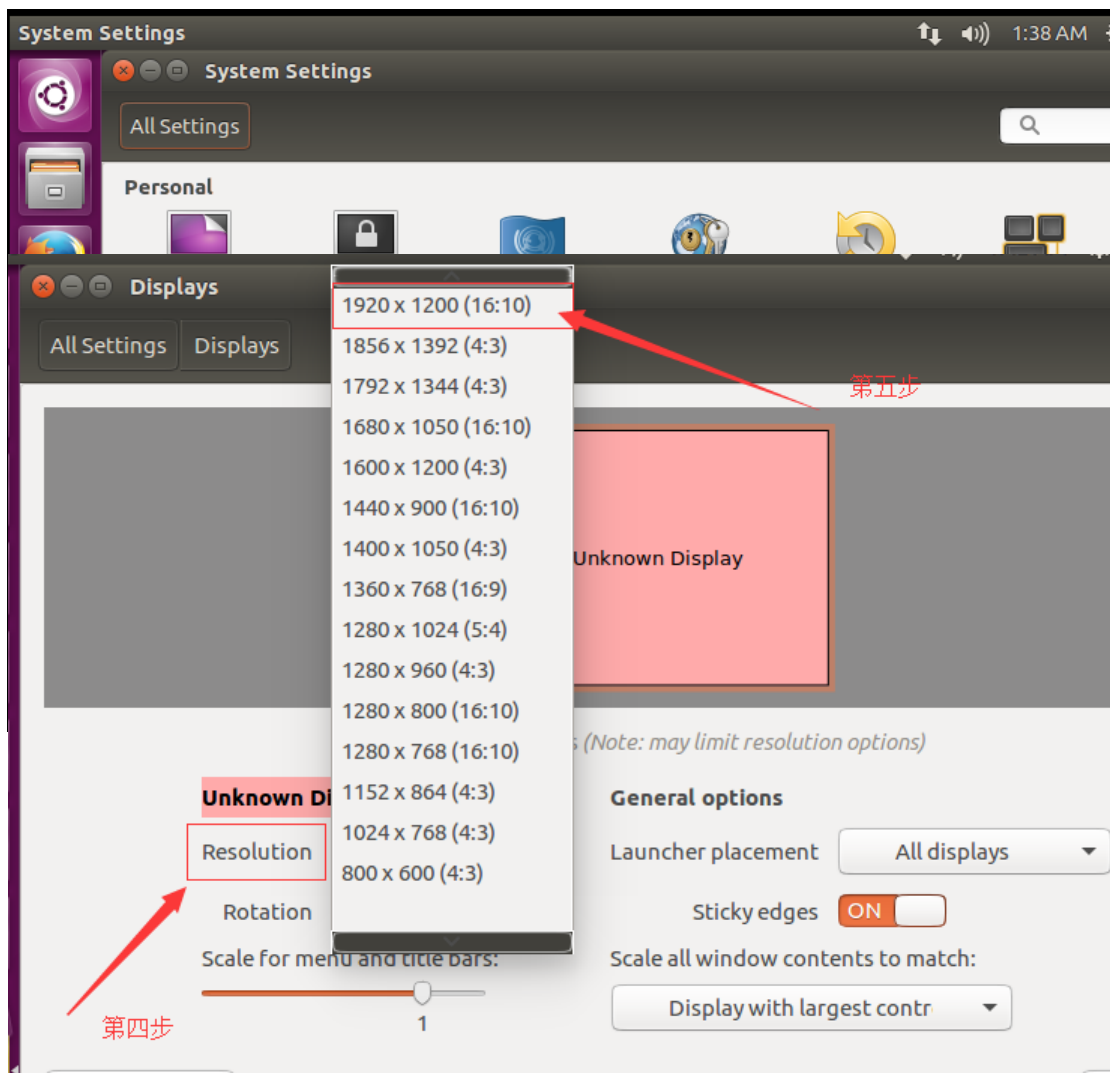
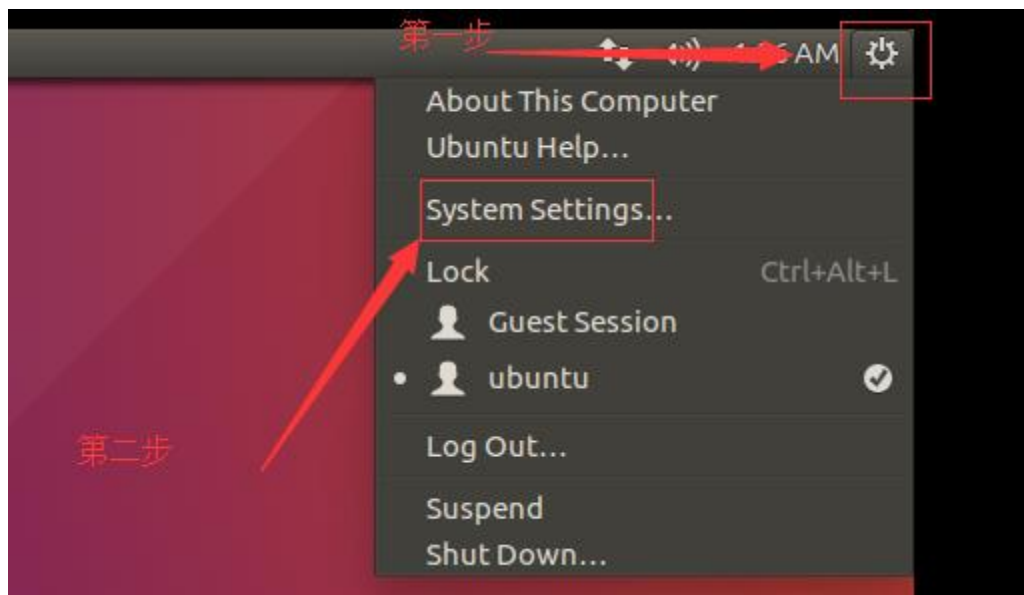
1.1.1 问题 1

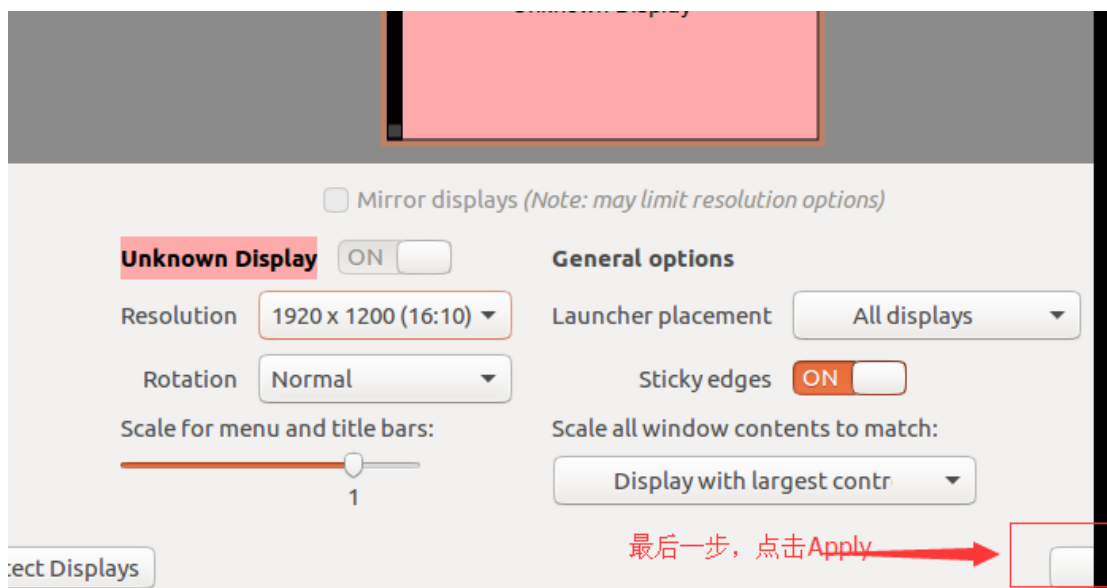
Q1: 在 VMware 中安装桌面版 ubuntu 成功后，进入桌面后，桌面未铺满 VMware 窗口？形如下图：



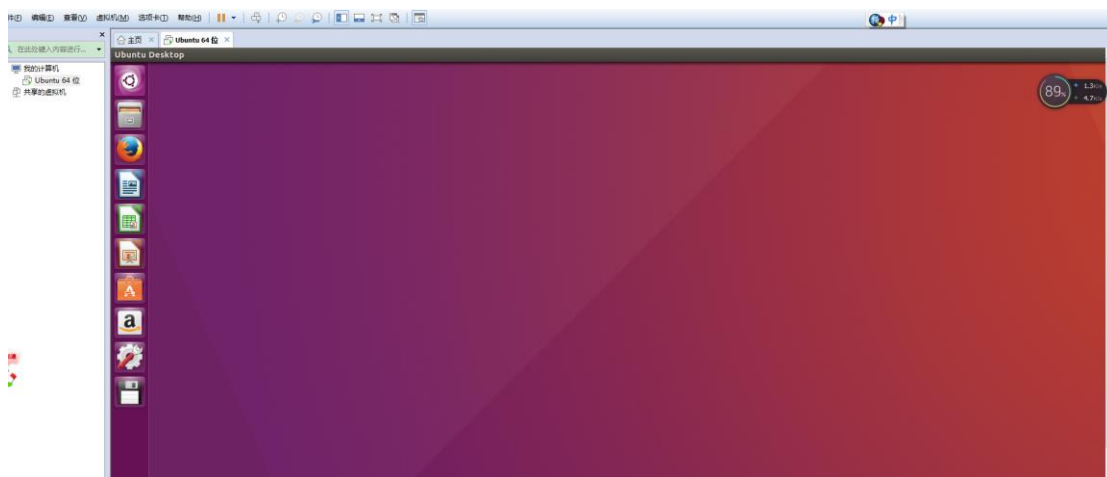
A: 网友的回答是：如果是服务器版是没法调大的，如果是桌面版，设置虚拟机里 ubuntu 的分辨率就可以了。亲测成功！（另一种建议是 vmware tools，这个没有测试过！）那怎么调分辨率呢？

很简单！





这是搞定之后的效果！就要这种效果！



1.1.2 问题 2

Q2: 我把截图粘贴到 Word2013 上，为什么只显示最底下一部分？

A: 网友的回答：出现此故障的原因，有可能是设置为固定值的文档行距小于图形的高度，从而导致插入的图形只显示出了一部分。所以要调整图片的段落格式中的行间距。方法是这样的： 1、点击图片后； 2、选择 Word 菜单，“格式”，“段落”3、在弹出的“段落”对话框中，将行距的，“固定值”改为“单倍行距”，即可。（亲测有效！）

so far,so good! 开始干活!

1.2 修改 apt 源

三步曲!

1、首先, 备份一下 ubuntu 原来的源地址列表文件

```
sudo cp /etc/apt/sources.list /etc/apt/sources.list.old
```

2、修改远地址。然后进行修改(手动修改)

```
sudo gedit /etc/apt/sources.list
```

可以在里面添加资源地址。

```
sudo sed -i 's/us/cn/g' /etc/apt/sources.list (命令自动修改)
```

3、更新

```
sudo apt-get update
```

修改 apt-get 源教程:

<http://jingyan.baidu.com/article/b2c186c82a2952c46ef6ffe1.html>

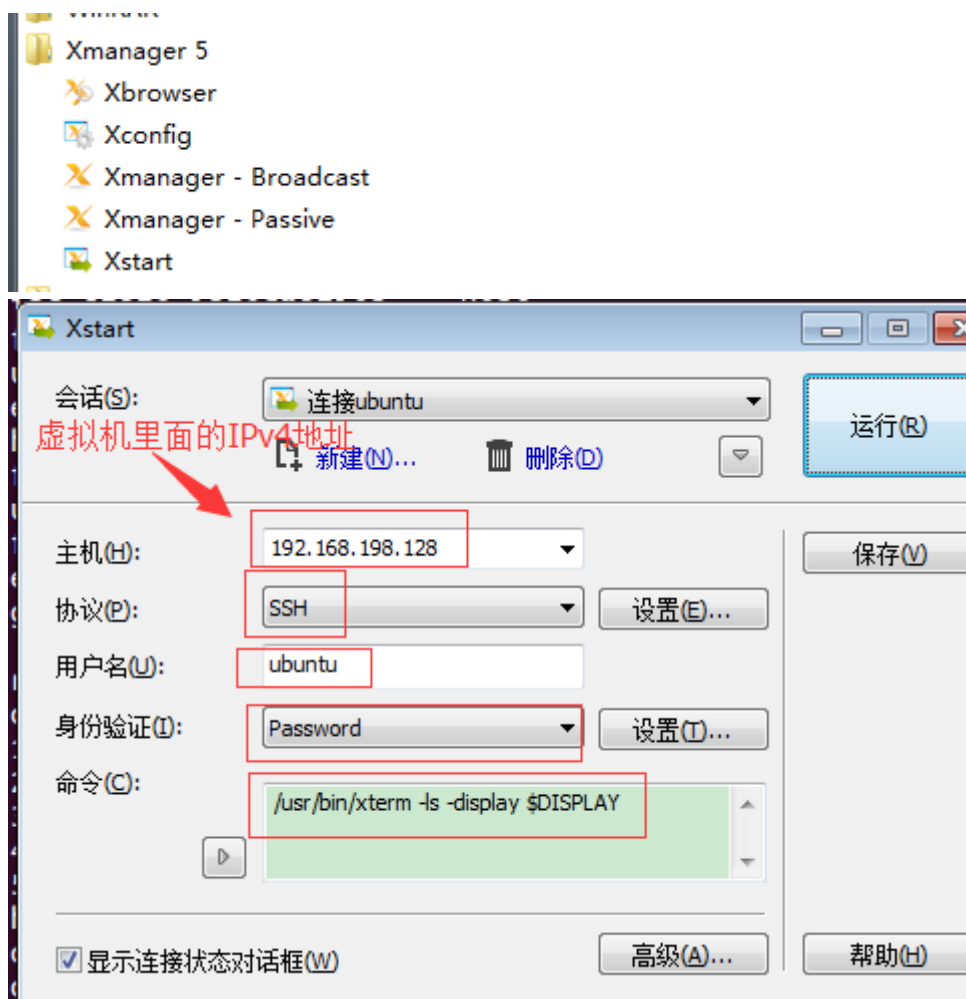
这一步非常关键!!! 后面需要大量的开源软件安装, 理论上可以节省数小时甚至数天。

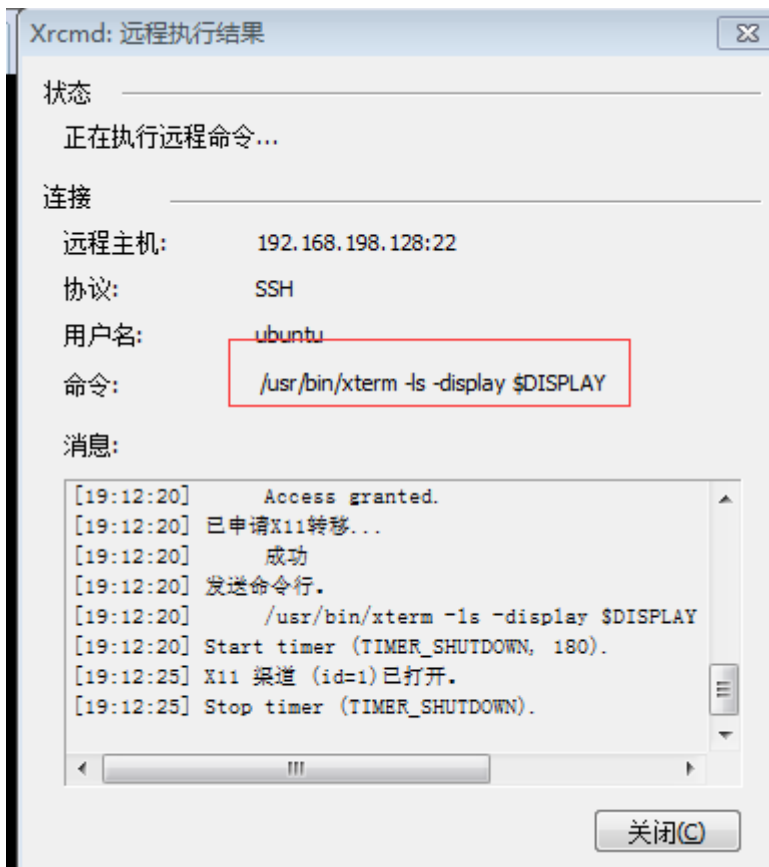
1.3 Xmanager 连接 ubuntu

xshell 收费! 我们用 xmanager!

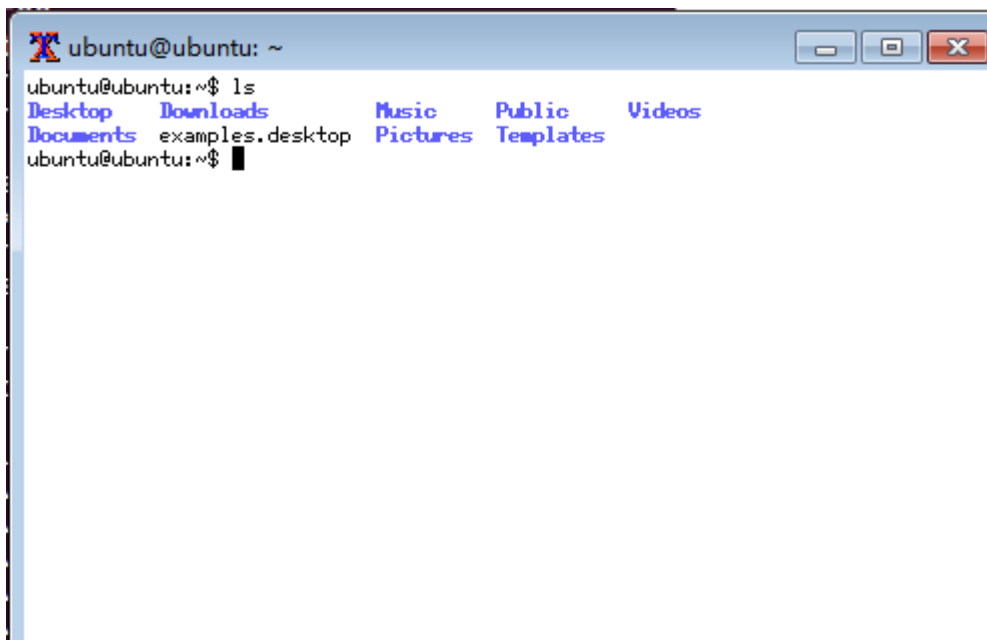
Xmanager 是一个简单易用的高性能的运行在 Windows 平台上的 X Server 软件。它能把远端 Unix/Linux 的桌面无缝地带到你的 Windows 上, 甚至当你的电脑是在内网或防火墙后, 你也能通过 SSH 协议安全的运行远端的 X 应用程序。特点包括: -可通过 Xconfig 工具设置多个 Xmanager 设置; -支持多用户的 Windows 终端环境; -支持多个 IP 地址; -支持本地资源数据库; -通过热键转换键盘映射; -支持多窗口下的 Windows 打印功能等。

安装完的目录:





生成的模拟终端:



注: xterm 是终端模拟器

小贴士: 模拟终端复制: Ctrl+Insert; 粘贴: Shift+Insert

(下面是 Xshell 连接 ubuntu 的一些思路。未亲测!)

首先，判断 Ubuntu 是否安装了 ssh 服务：

1、输入：`#ps -e |grep ssh` 如果服务已经启动，则可以同时看到“ssh-agent”和“sshd”，否则表示没有安装服务，或没有开机启动

2、安装 ssh 服务，输入命令：`#sudo apt-get install openssh-server`

3、启动服务：`#!/etc/init.d/ssh start`

4、本机测试是否能够成功登录：`#ssh -l 用户名 本机 ip`

Xshell 连接 ubuntu 教程：

<http://blog.csdn.net/qingyuanluofeng/article/details/44099863>

)

1.4 安装基础软件和工具

1.4.1 安装 cURL

命令：`sudo apt-get install curl`

1.4.2 安装 docker 和 docker compose

1、安装 docker 命令：`# curl -sSL https://get.daocloud.io/docker | sh`

建议安装最新版本的 Docker。至少是 v1.12 及更高版本。

2、检查安装的 docker 版本，命令：`docker -version`

3、启动 docker 服务命令：`sudo service docker start`

（如果所有都没问题，这里不会有输出！）

4、查看 docker 的命令：`docker -help`

5、添加用户到 docker group 中：

如果还没有 docker group 就添加一个: `sudo groupadd docker`

将该用户加入该 group 内。然后退出并重新登录就生效了。

`sudo gpasswd -a ${USER} docker`

重启 docker 服务: `sudo service docker restart`

安装 compose 之前,要先安装 docker!

安装 compose 命令: `sudo apt install python-pip` (第一步安装 pip)

`sudo pip install docker-compose` (第二步安装 docker-compose)

测试 compose 版本: `docker-compose --version`

建议安装最新版本的 Docker-compose。至少是 v1.8 及更高版本。

1.5 安装 Go

Hyperledger Fabric 的许多组件都使用了 Go 语言 1.7.X 的特性。

下载 Go 安装包 `go1.7.6.linux-amd64.tar.gz`

1、解压缩

`tar zxvf go1.2.2.linux-amd64.tar.gz`

2、移动(要有 ROOT 权限)

`mv go /usr/local/go`

3、配置环境变量 (要重视, 否则到时候各种毛病)

以下全部都需要 ROOT 权限

`vi /etc/profile`

`export GOROOT=/usr/local/go` #设置为 go 安装的路径

`export GOPATH=$HOME/go` #默认安装包的路径


```
export PATH=$PATH:$GOROOT/bin:$GOPATH/bin
```

```
source /etc/profile
```

1.6 安装 Node.js 和 NPM

如果你即将使用 Node.js 版的 SDK 开发应用,你得安装版本 6.9.x 的 Node.js。

安装 node.js 会附带把 npm 安好,但建议查看一下 npm 的版本。

升级 npm 工具命令: `npm install npm@3.10.10 -g`

nvm (node version manager)

配置环境变量:

```
export NODE_HOME=/home/*****/.nvm/versions/node/v6.9.5/bin
```

(*处为虚拟机用户名)

```
export PATH=$PATH:$NODE_HOME/bin
```

```
export NODE_PATH=$NODE_HOME/lib/node_modules
```

```
source /etc/profile
```

查看版本: `node -v`

```
npm -v
```

第二章 Getting started:入门指南

2.1 安装准备

见第一章

2.2 安装 fabric 二进制文件和相应的 docker 镜像

见本章 2.3.1 节

2.3 获取 Fabric 样例

首先**创建**一个目录存放 Fabric 样例，接着在该目录下打开终端。

- 1、 创建目录命令：

```
mkdir -p $GOPATH/src/github.com/hyperledger/
```

- 2、 切入刚创建的目录下以及克隆样例

```
cd $GOPATH/src/github.com/hyperledger/
```

```
git clone https://github.com/hyperledger/fabric-samples.git
```

- 3、 切进 fabric-samples 目录：cd fabric-samples

2.3.1 下载特定平台的二进制文件

安装特定平台的 Fabric 二进制文件，这个过程是辅助上述 Fabric 样例，但可以独立使用。如果你不安装上述样例，那么可以创建一个目录，并 cd 该目录，提取特定平台的二进制文件。

命令：curl -sSL <https://goo.gl/Gci9ZX> | bash

上面的命令会下载以及执行一个 bash 脚本（脚本的功能就是从 Docker Hub 下载特定平台特定版本的 fabric docker 镜像并将它们自动标记为 latest）。脚本会取回四个特定平台的二进制文件（就是可执行文件）。他们分别是：cryptogen, configtxgen, configtxlator, and peer。他们都存放在当前工作目录

（\$GOPATH/src/github.com/hyperledger/fabric-samples）的子目录 bin 下。

执行完命令后，在 bin 目录可以看到这些可执行文件。

```
configtxgen configtxlator cryptogen get-byfn.sh get-docker-images.sh orderer peer
```

将它们添加到 PATH 中去。（添加 PATH 的方法见第一章 1.5 节）

```
export
```

```
PATH=$PATH:$GOROOT/bin:$GOPATH/bin:$GOPATH/src/github.com
```

/hyperledger/fabric-samples/bin

这是脚本具体内容:

```
#!/bin/bash
#
# Copyright IBM Corp. All Rights Reserved.
#
# SPDX-License-Identifier: Apache-2.0
#

export VERSION=1.0.2
export ARCH=$(echo "$(uname -s|tr '[:upper:]' '[:lower:]'|sed
's/mingw64_nt.*\/windows/)-$(uname -m | sed 's/x86_64/amd64/g')" | awk '{print
tolower($0)}')
#Set MARCH variable i.e ppc64le,s390x,x86_64,i386
MARCH=`uname -m`

dockerFabricPull() {
    local FABRIC_TAG=$1
    for IMAGES in peer orderer couchdb ccenv javaenv kafka zookeeper tools; do
        echo "==> FABRIC IMAGE: $IMAGES"
        echo
        docker pull hyperledger/fabric-$IMAGES:$FABRIC_TAG
        docker tag hyperledger/fabric-$IMAGES:$FABRIC_TAG
hyperledger/fabric-$IMAGES
    done
}

dockerCaPull() {
    local CA_TAG=$1
    echo "==> FABRIC CA IMAGE"
    echo
    docker pull hyperledger/fabric-ca:$CA_TAG
    docker tag hyperledger/fabric-ca:$CA_TAG hyperledger/fabric-ca
}

: ${CA_TAG:="$MARCH-$VERSION"}
: ${FABRIC_TAG:="$MARCH-$VERSION"}

echo "===> Downloading platform binaries"
curl
https://nexus.hyperledger.org/content/repositories/releases/org/hyperledger/fabric/hyp
```

```
erledger-fabric/${ARCH}-${VERSION}/hyperledger-fabric-${ARCH}-  
${VERSION}.tar.gz | tar xz
```

```
echo "====> Pulling fabric Images"  
dockerFabricPull ${FABRIC_TAG}  
  
echo "====> Pulling fabric ca Image"  
dockerCaPull ${CA_TAG}  
echo  
echo "====> List out hyperledger docker images"  
docker images | grep hyperledger*
```

如果上面的命令执行完出错。比如我在敲完上述命令时，返回了错误“curl: (7) Failed to connect to goo.gl port 443: Connection refused”。应该是网络的原因！

那么还有一种办法：手动下载这个脚本执行。具体怎么弄呢？

- 1、 用 vi 编辑器创建一个文件（例如：vi dockerFabricPull.sh），将
<https://raw.githubusercontent.com/hyperledger/fabric/master/scripts/bootstrap-1.0.2.sh> 的脚本复制过去，保存！
- 2、 给这个文件赋以可执行权限：
chmod +x ./dockerFabricPull.sh
- 3、 执行： ./ ./dockerFabricPull.sh

上述三步我是在 root 权限下搞定的！

下面是第三步执行的输出：

```

root@ubuntu:/home/ubuntu/go/src/github.com/hyperledger/fabric-samples# chmod +x ./dockerFabricPull.sh
root@ubuntu:/home/ubuntu/go/src/github.com/hyperledger/fabric-samples# ./dockerFabricPull.sh
==> Downloading platform binaries
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left   Speed
100 22.6M  100 22.6M    0     0   207k      0  0:01:51  0:01:51 --:--:-- 242k
==> Pulling fabric Images
==> FABRIC IMAGE: peer

x86_64-1.0.2: Pulling from hyperledger/fabric-peer
d5c6f90da05d: Pull complete
1300883d87d5: Pull complete
c220aa3cfc1b: Pull complete
2e9398f099dc: Pull complete
dc27a084064f: Pull complete
87675a6d4030: Pull complete
93e601aafda8: Pull complete
51fc526ed766: Pull complete
e0386e9ee7cf: Pull complete
a75d8afcee28: Pull complete
Digest: sha256:ce6e3c3d0f71641020184e4fe0615c2e65b5181664bad3d05942091917b73e28
Status: Downloaded newer image for hyperledger/fabric-peer:x86_64-1.0.2
==> FABRIC IMAGE: orderer

x86_64-1.0.2: Pulling from hyperledger/fabric-orderer
d5c6f90da05d: Already exists
1300883d87d5: Already exists
c220aa3cfc1b: Already exists
2e9398f099dc: Already exists
dc27a084064f: Already exists
87675a6d4030: Already exists
93e601aafda8: Already exists
51fc526ed766: Already exists
f5aa61fc7717: Pull complete
ebcf07becb40: Pull complete
Digest: sha256:898f6e0fae6bcdaf732a8c58779471b6852ce4fb21ffa5dc495cb8ada16639
Status: Downloaded newer image for hyperledger/fabric-orderer:x86_64-1.0.2
==> FABRIC IMAGE: couchdb

x86_64-1.0.2: Pulling from hyperledger/fabric-couchdb
d5c6f90da05d: Already exists
1300883d87d5: Already exists
c220aa3cfc1b: Already exists
2e9398f099dc: Already exists
dc27a084064f: Already exists
87675a6d4030: Already exists
93e601aafda8: Already exists
6be6b5cbce6b: Pull complete
c90d7a44426d: Pull complete
9767b1c8440b: Pull complete
b9b680c73863: Downloading [=====] 238.1MB/253.4MB
910808d7b46d: Download complete
b2b400af2b65: Download complete
656c0e3d1026: Download complete
fd537a07cc88: Download complete
fb80790f85e2: Download complete
6b4be879a573: Download complete
24e26e38b19c: Download complete
ba4d4ddd242e: Download complete
■

```

下载完列出的 docker 镜像:

```

==> List out hyperledger docker images
hyperledger/fabric-ca          latest          7e53be14f804   5 days ago     238MB
hyperledger/fabric-ca          x86_64-1.0.2  7e53be14f804   5 days ago     238MB
hyperledger/fabric-tools       latest          ba9750b2565d   5 days ago     1.33GB
hyperledger/fabric-tools       x86_64-1.0.2  ba9750b2565d   5 days ago     1.33GB
hyperledger/fabric-couchdb     latest          3f922f54bd68   5 days ago     1.47GB
hyperledger/fabric-couchdb     x86_64-1.0.2  3f922f54bd68   5 days ago     1.47GB
hyperledger/fabric-kafka       latest          0b4b1d249e65   5 days ago     1.29GB
hyperledger/fabric-kafka       x86_64-1.0.2  0b4b1d249e65   5 days ago     1.29GB
hyperledger/fabric-zookeeper   latest          1efb063147d3   5 days ago     1.3GB
hyperledger/fabric-zookeeper   x86_64-1.0.2  1efb063147d3   5 days ago     1.3GB
hyperledger/fabric-orderer     latest          6efd17e86e65   5 days ago     151MB
hyperledger/fabric-orderer     x86_64-1.0.2  6efd17e86e65   5 days ago     151MB
hyperledger/fabric-peer        latest          0e2ed51971c9   5 days ago     154MB
hyperledger/fabric-peer        x86_64-1.0.2  0e2ed51971c9   5 days ago     154MB
hyperledger/fabric-javaenv     latest          0be45dbd7ff4   5 days ago     1.41GB
hyperledger/fabric-javaenv     x86_64-1.0.2  0be45dbd7ff4   5 days ago     1.41GB
hyperledger/fabric-ocenv       latest          d0f166e1a89e   5 days ago     1.28GB
hyperledger/fabric-ocenv       x86_64-1.0.2  d0f166e1a89e   5 days ago     1.28GB
root@ubuntu:/home/ubuntu/go/src/github.com/hyperledger/fabric-samples# █

```

这些所有镜像构成了 Hyperledger Fabric 网络。

2.4 API 文档

<https://godoc.org/github.com/hyperledger/fabric>

2.5 Fabric SDK

目前交付使用的是 SDK 是 Node.js 和 Java SDKs。1.0 之后，还将提供 python 和 Go 的 SDK。

Node.js SDK 文档：

<https://fabric-sdk-node.github.io/>

Java SDK 文档：<https://github.com/hyperledger/fabric-sdk-java>

2.6 Fabric CA

Fabric 提供了可选的证书机构服务。用来生成证书和密钥材料。

2.7 教程

为了开启你的 Fabric 之旅，fabric 官方给了四个教程。第一个是面向 Fabric 应用程序开发者，它将全程带你使用 NodeSDK 写首个 fabric 应用。第二是面向 Fabric 运维人员。最后两个是链上代码教程，一个

面向开发者，一个面向运维人员。